

settling time, and peak time of the closed-loop system for a step input; (d) calculate the step response for the closed-loop system; and (e) for the simplified model, design the system gain to meet a transient response requirement.

- Given the block diagrams for the Unmanned Free-Swimming Submersible (UFSS) vehicle's pitch and heading control systems on the back endpapers, you will be able to represent each control system in state space.

5.1 Introduction

We have been working with individual subsystems represented by a block with its input and output. More complicated systems, however, are represented by the interconnection of many subsystems. Since the response of a single transfer function can be calculated, we want to represent multiple subsystems as a single transfer function. We can then apply the analytical techniques of the previous chapters and obtain transient response information about the entire system.

In this chapter, multiple subsystems are represented in two ways: as block diagrams and as signal-flow graphs. Although neither representation is limited to a particular analysis and design technique, block diagrams are usually used for frequency-domain analysis and design, and signal-flow graphs for state-space analysis.

Signal-flow graphs represent transfer functions as lines, and signals as small-circular nodes. Summing is implicit. To show why it is convenient to use signal-flow graphs for state-space analysis and design, consider Figure 3.10. A graphical representation of a system's transfer function is as simple as Figure 3.10(a). However, a graphical representation of a system in state space requires representation of each state variable, as in Figure 3.10(b). In that example, a single-block transfer function requires seven blocks and a summing junction to show the state variables explicitly. Thus, signal-flow graphs have advantages over block diagrams, such as Figure 3.10(b): They can be drawn more quickly, they are more compact, and they emphasize the state variables.

We will develop techniques to reduce each representation to a single transfer function. Block diagram algebra will be used to reduce block diagrams and Mason's rule to reduce signal-flow graphs. Again, it must be emphasized that these methods are typically used as described. As we shall see, however, either method can be used for frequency-domain or state-space analysis and design.

5.2 Block Diagrams

As you already know, a subsystem is represented as a block with an input, an output, and a transfer function. Many systems are composed of multiple subsystems, as in Figure 5.1. When multiple subsystems are interconnected, a few more schematic elements must be added to the block diagram. These new elements are *summing junctions* and *pickoff points*. All component parts of a block diagram for a linear, time-invariant system are shown in Figure 5.2. The characteristic of the summing junction shown in Figure 5.2(c) is that the output signal, $C(s)$, is the algebraic sum of the input signals, $R_1(s)$, $R_2(s)$, and $R_3(s)$. The figure shows three inputs, but any

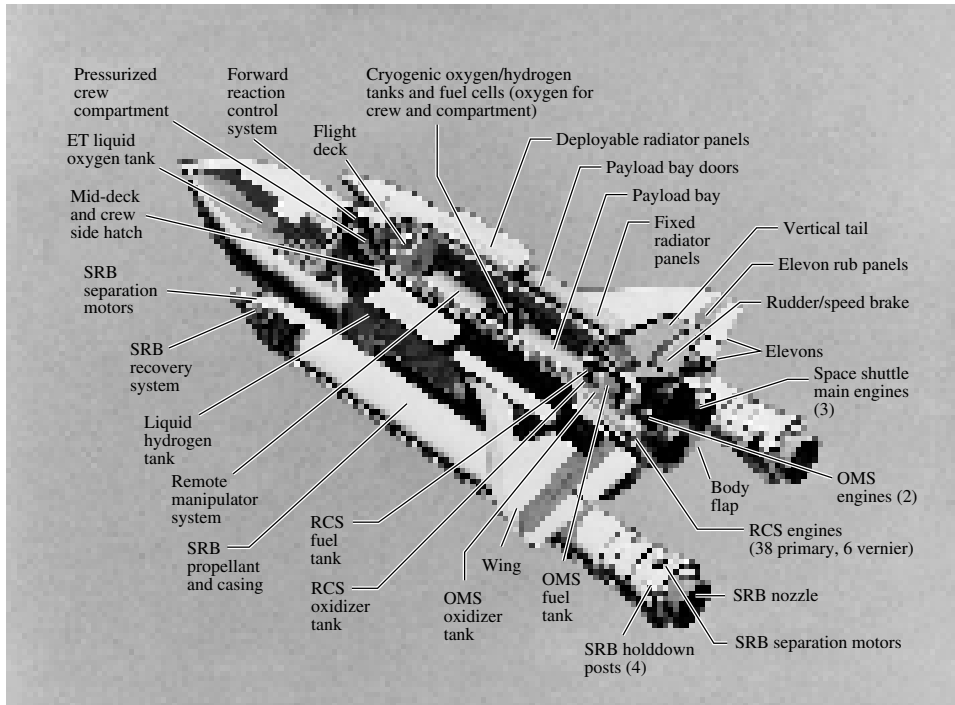


FIGURE 5.1 The space shuttle consists of multiple subsystems. Can you identify those that are control systems or parts of control systems?

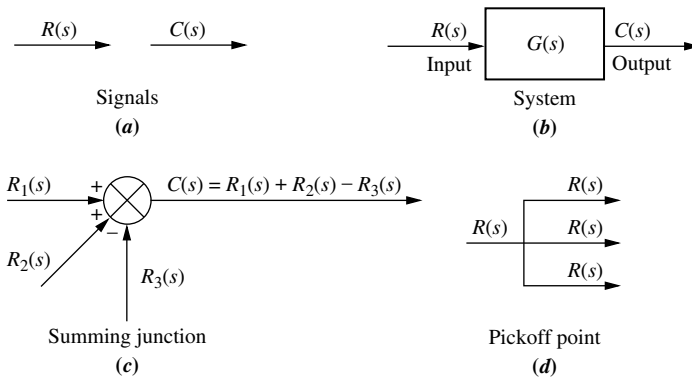


FIGURE 5.2 Components of a block diagram for a linear, time-invariant system

number can be present. A pickoff point, as shown in Figure 5.2(d), distributes the input signal, $R(s)$, undiminished, to several output points.

We will now examine some common topologies for interconnecting subsystems and derive the single transfer function representation for each of them. These common topologies will form the basis for reducing more complicated systems to a single block.

Cascade Form

Figure 5.3(a) shows an example of cascaded subsystems. Intermediate signal values are shown at the output of each subsystem. Each signal is derived from the product of the input times the transfer function. The equivalent transfer function, $G_e(s)$, shown in Figure 5.3(b), is the output Laplace transform divided by the input Laplace

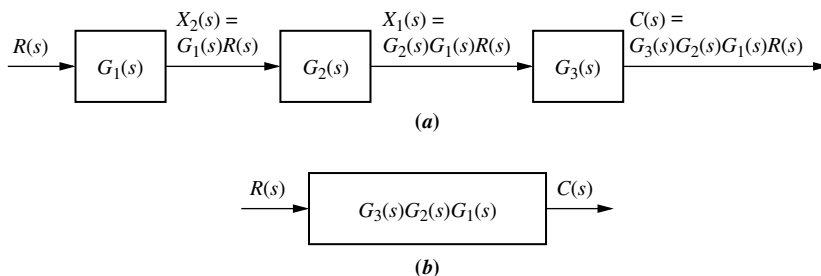


FIGURE 5.3 a. Cascaded subsystems; b. equivalent transfer function

transform from Figure 5.3(a), or

$$G_e(s) = G_3(s)G_2(s)G_1(s) \quad (5.1)$$

which is the product of the subsystems' transfer functions.

Equation (5.1) was derived under the assumption that interconnected subsystems do not load adjacent subsystems. That is, a subsystem's output remains the same whether or not the subsequent subsystem is connected. If there is a change in the output, the subsequent subsystem loads the previous subsystem, and the equivalent transfer function is not the product of the individual transfer functions. The network of Figure 5.4(a) demonstrates this concept. Its transfer function is

$$G_1(s) = \frac{V_1(s)}{V_i(s)} = \frac{1}{s + \frac{R_1 C_1}{R_1 C_1}} \quad (5.2)$$

Similarly, the network of Figure 5.4(b) has the following transfer function:

$$G_2(s) = \frac{V_2(s)}{V_1(s)} = \frac{1}{s + \frac{R_2 C_2}{R_2 C_2}} \quad (5.3)$$

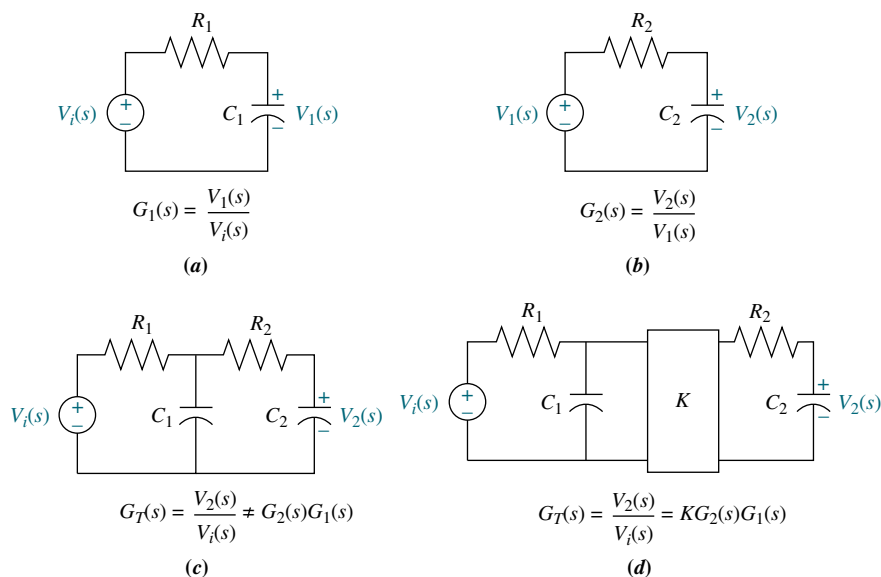


FIGURE 5.4 Loading in cascaded systems

If the networks are placed in cascade, as in Figure 5.4(c), you can verify that the transfer function found using loop or node equations is

$$G(s) = \frac{V_2(s)}{V_i(s)} = \frac{1}{s^2 + \left(\frac{1}{R_1 C_1} + \frac{1}{R_2 C_2} + \frac{1}{R_2 C_1} \right) s + \frac{1}{R_1 C_1 R_2 C_2}} \quad (5.4)$$

But, using Eq. (5.1),

$$G(s) = G_2(s)G_1(s) = \frac{1}{s^2 + \left(\frac{1}{R_1 C_1} + \frac{1}{R_2 C_2} \right) s + \frac{1}{R_1 C_1 R_2 C_2}} \quad (5.5)$$

Equations (5.4) and (5.5) are not the same: Eq. (5.4) has one more term for the coefficient of s in the denominator and is correct.

One way to prevent loading is to use an amplifier between the two networks, as shown in Figure 5.4(d). The amplifier has a high-impedance input, so that it does not load the previous network. At the same time it has a low-impedance output, so that it looks like a pure voltage source to the subsequent network. With the amplifier included, the equivalent transfer function is the product of the transfer functions and the gain, K , of the amplifier.

Parallel Form

Figure 5.5 shows an example of parallel subsystems. Again, by writing the output of each subsystem, we can find the equivalent transfer function. Parallel subsystems have a common input and an output formed by the algebraic sum of the outputs from all of the subsystems. The equivalent transfer function, $G_e(s)$, is the output transform divided by the input transform from Figure 5.5(a), or

$$G_e(s) = \pm G_1(s) \pm G_2(s) \pm G_3(s) \quad (5.6)$$

which is the algebraic sum of the subsystems' transfer functions; it appears in Figure 5.5(b).

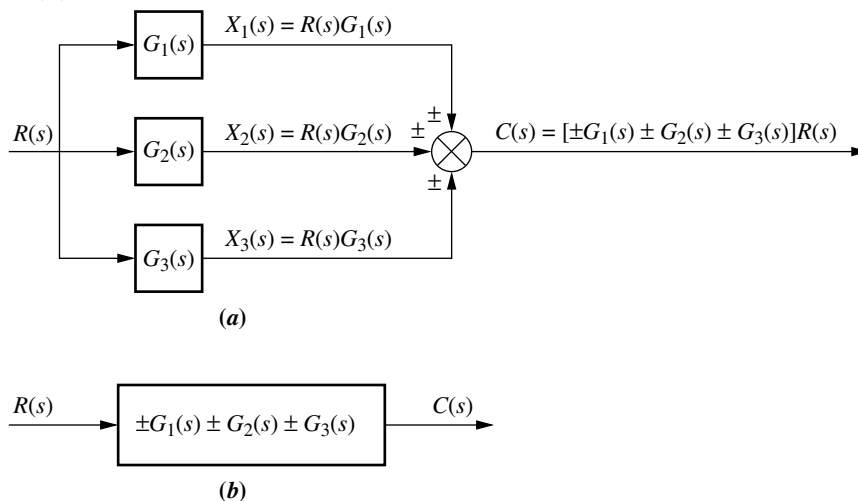


FIGURE 5.5 **a.** Parallel subsystems; **b.** equivalent transfer function

Feedback Form

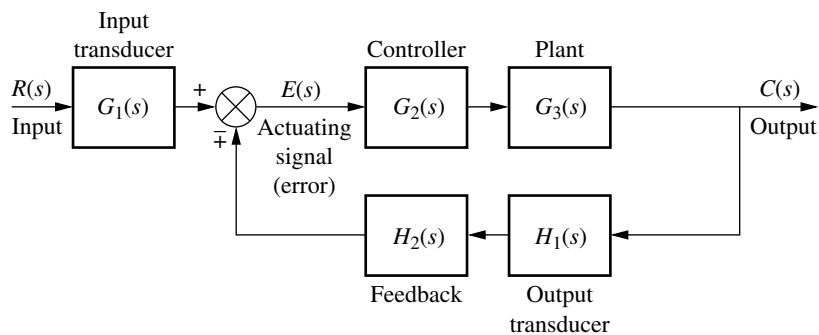
The third topology is the feedback form, which will be seen repeatedly in subsequent chapters. The feedback system forms the basis for our study of control systems engineering. In Chapter 1, we defined open-loop and closed-loop systems and pointed out the advantage of closed-loop, or feedback control, systems over open-loop systems. As we move ahead, we will focus on the analysis and design of feedback systems.

Let us derive the transfer function that represents the system from its input to its output. The typical feedback system, described in detail in Chapter 1, is shown in Figure 5.6(a); a simplified model is shown in Figure 5.6(b).¹ Directing our attention to the simplified model,

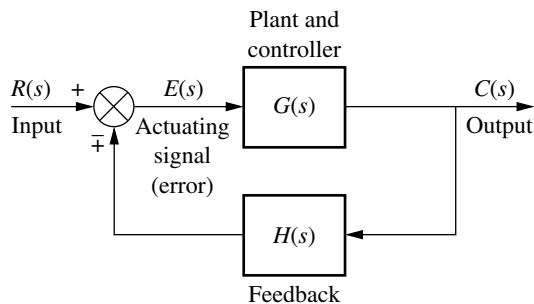
$$E(s) = R(s) \mp C(s)H(s) \quad (5.7)$$

But since $C(s) = E(s)G(s)$,

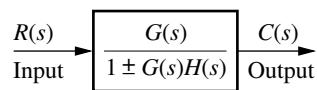
$$E(s) = \frac{C(s)}{G(s)} \quad (5.8)$$



(a)



(b)



(c)

FIGURE 5.6 a. Feedback control system; b. simplified model; c. equivalent transfer function

¹The system is said to have *negative feedback* if the sign at the summing junction is negative and *positive feedback* if the sign is positive.

Substituting Eq. (5.8) into Eq. (5.7) and solving for the transfer function, $C(s)/R(s) = G_e(s)$, we obtain the equivalent, or *closed-loop*, transfer function shown in Figure 5.6(c),

$$G_e(s) = \frac{G(s)}{1 \pm G(s)H(s)} \quad (5.9)$$

The product, $G(s)H(s)$, in Eq. (5.9) is called the *open-loop transfer function*, or *loop gain*.

So far, we have explored three different configurations for multiple subsystems. For each, we found the equivalent transfer function. Since these three forms are combined into complex arrangements in physical systems, recognizing these topologies is a prerequisite to obtaining the equivalent transfer function of a complex system. In this section, we will reduce complex systems composed of multiple subsystems to single transfer functions.

Moving Blocks to Create Familiar Forms

Before we begin to reduce block diagrams, it must be explained that the familiar forms (cascade, parallel, and feedback) are not always apparent in a block diagram. For example, in the feedback form, if there is a pickoff point after the summing junction, you cannot use the feedback formula to reduce the feedback system to a single block. That signal disappears, and there is no place to reestablish the pickoff point.

This subsection will discuss basic block moves that can be made to order to establish familiar forms when they almost exist. In particular, it will explain how to move blocks left and right past summing junctions and pickoff points.

Figure 5.7 shows equivalent block diagrams formed when transfer functions are moved left or right past a summing junction, and Figure 5.8 shows equivalent block diagrams formed when transfer functions are moved left or right past a pickoff point. In the diagrams the symbol \equiv means “equivalent to.” These equivalences,

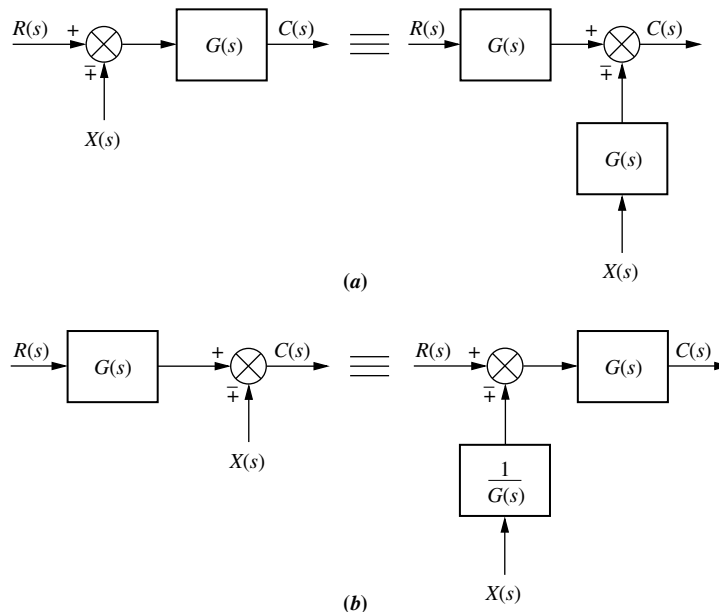


FIGURE 5.7 Block diagram algebra for summing junctions—equivalent forms for moving a block **a.** to the left past a summing junction; **b.** to the right past a summing junction

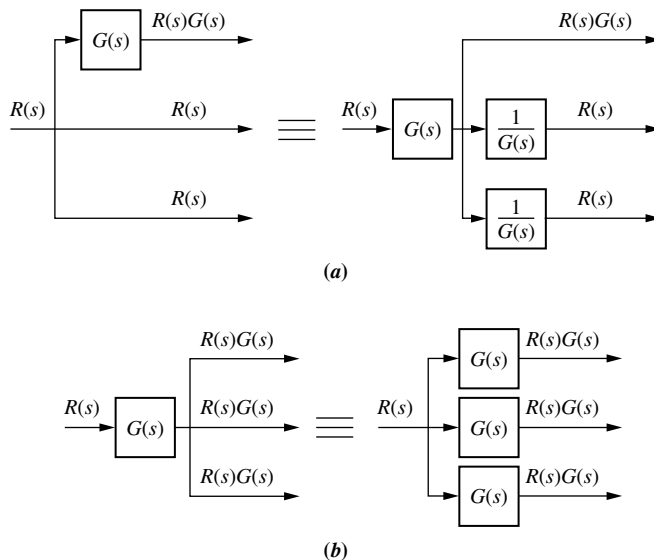


FIGURE 5.8 Block diagram algebra for pickoff points—equivalent forms for moving a block **a.** to the left past a pickoff point; **b.** to the right past a pickoff point

along with the forms studied earlier in this section, can be used to reduce a block diagram to a single transfer function. In each case of Figures 5.7 and 5.8, the equivalence can be verified by tracing the signals at the input through to the output and recognizing that the output signals are identical. For example, in Figure 5.7(a), signals $R(s)$ and $X(s)$ are multiplied by $G(s)$ before reaching the output. Hence, both block diagrams are equivalent, with $C(s) = R(s)G(s) \mp X(s)G(s)$. In Figure 5.7(b), $R(s)$ is multiplied by $G(s)$ before reaching the output, but $X(s)$ is not. Hence, both block diagrams in Figure 5.7(b) are equivalent, with $C(s) = R(s)G(s) \mp X(s)$. For pickoff points, similar reasoning yields similar results for the block diagrams of Figure 5.8(a) and (b).

Let us now put the whole story together with examples of block diagram reduction.

Example 5.1

Block Diagram Reduction via Familiar Forms

PROBLEM: Reduce the block diagram shown in Figure 5.9 to a single transfer function.

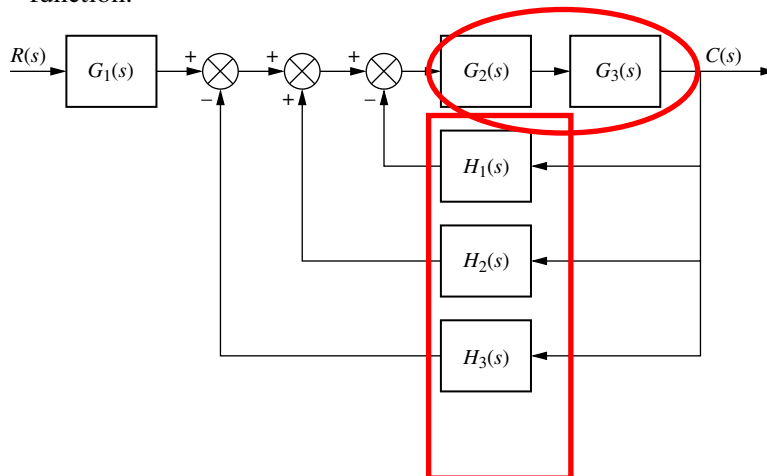


FIGURE 5.9 Block diagram for Example 5.1

SOLUTION: We solve the problem by following the steps in Figure 5.10. First, the three summing junctions can be collapsed into a single summing junction, as shown in Figure 5.10(a).

Second, recognize that the three feedback functions, $H_1(s)$, $H_2(s)$, and $H_3(s)$, are connected in parallel. They are fed from a common signal source, and their outputs are summed. The equivalent function is $H_1(s) - H_2(s) + H_3(s)$. Also recognize that $G_2(s)$ and $G_3(s)$ are connected in cascade. Thus, the equivalent transfer function is the product, $G_3(s)G_2(s)$. The results of these steps are shown in Figure 5.10(b).

Finally, the feedback system is reduced and multiplied by $G_1(s)$ to yield the equivalent transfer function shown in Figure 5.10(c).

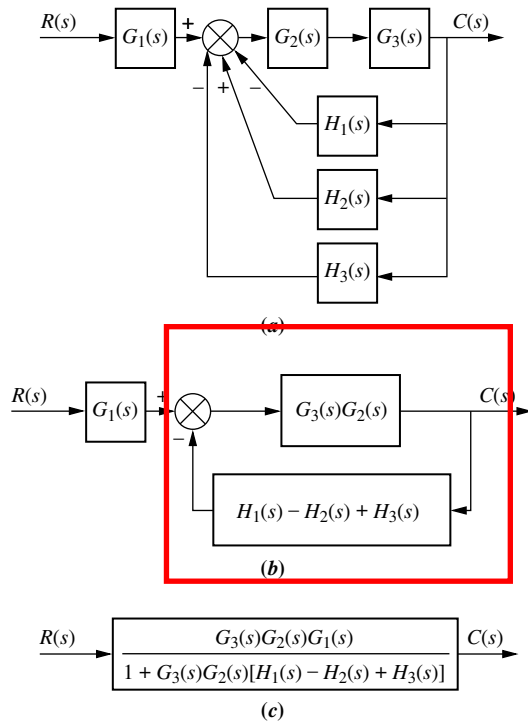


FIGURE 5.10 Steps in solving Example 5.1: **a.** Collapse summing junctions; **b.** form equivalent cascaded system in the forward path and equivalent parallel system in the feedback path; **c.** form equivalent feedback system and multiply by cascaded $G_1(s)$

Example 5.2

Block Diagram Reduction by Moving Blocks

PROBLEM: Reduce the system shown in Figure 5.11 to a single transfer function.

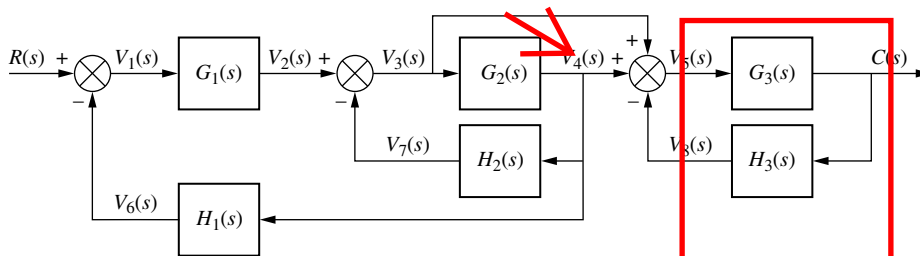


FIGURE 5.11 Block diagram for Example 5.2

SOLUTION: In this example we make use of the equivalent forms shown in Figures 5.7 and 5.8. First, move $G_2(s)$ to the left past the pickoff point to create parallel subsystems, and reduce the feedback system consisting of $G_3(s)$ and $H_3(s)$. This result is shown in Figure 5.12(a).

Second, reduce the parallel pair consisting of $1/G_2(s)$ and unity, and push $G_1(s)$ to the right past the summing junction, creating parallel subsystems in the feedback. These results are shown in Figure 5.12(b).

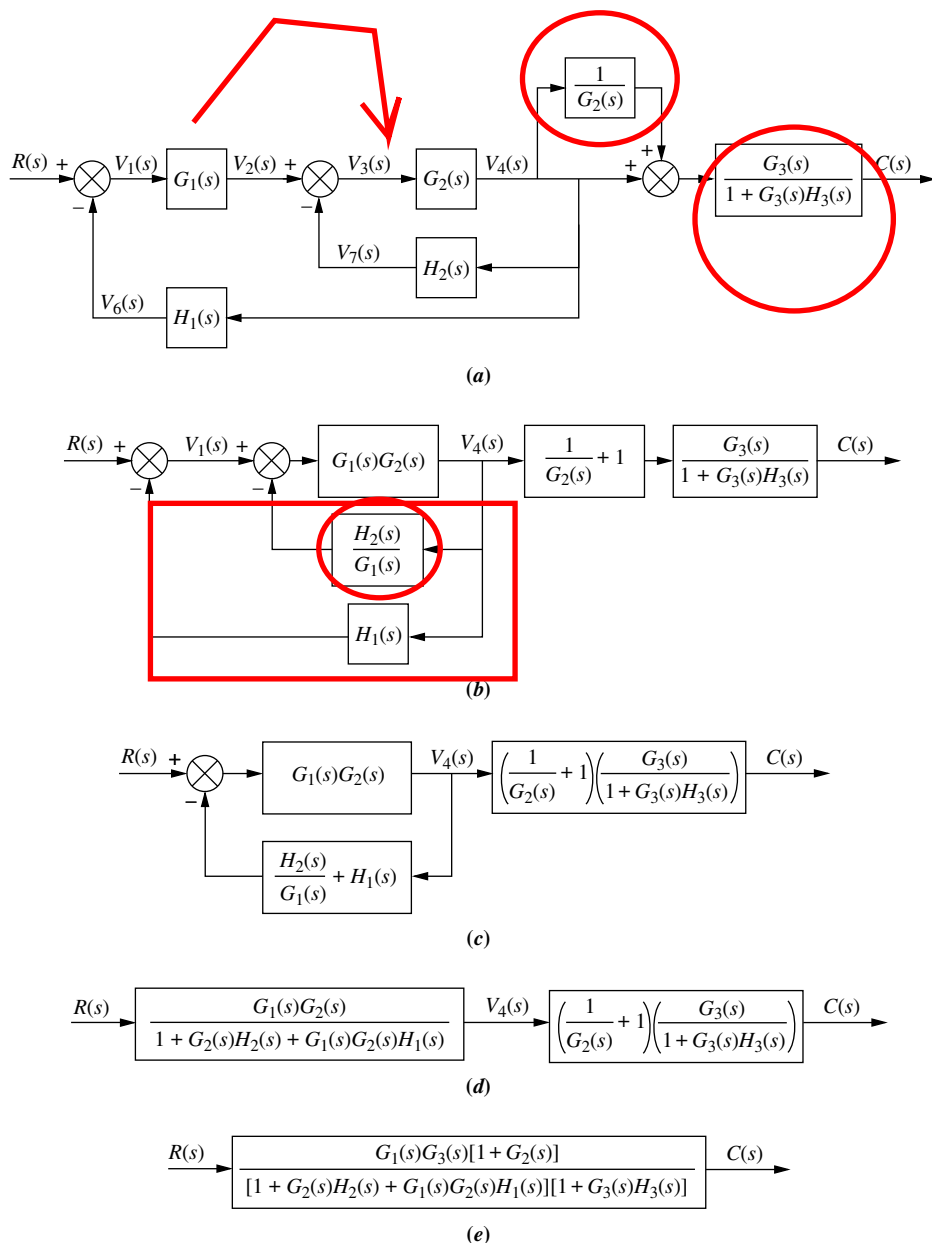


FIGURE 5.12 Steps in the block diagram reduction for Example 5.2

Third, collapse the summing junctions, add the two feedback elements together, and combine the last two cascaded blocks. Figure 5.12(c) shows these results. Fourth, use the feedback formula to obtain Figure 5.12(d). Finally, multiply the two cascaded blocks and obtain the final result, shown in Figure 5.12(e).

Students who are using MATLAB should now run ch5p1 in Appendix B to perform block diagram reduction.

MATLAB
ML

Skill-Assessment Exercise 5.1

PROBLEM: Find the equivalent transfer function, $T(s) = C(s)/R(s)$, for the system shown in Figure 5.13.

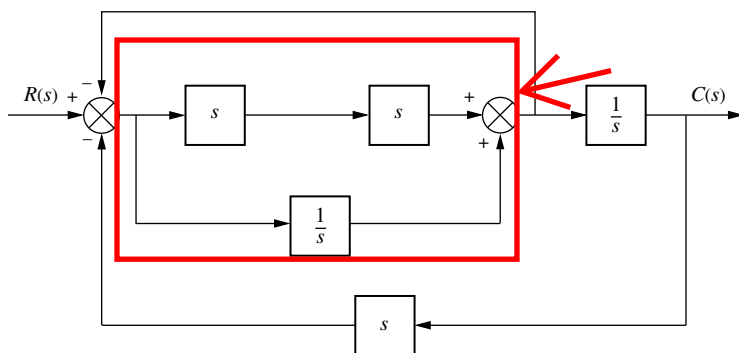


FIGURE 5.13 Block diagram for Skill-Assessment Exercise 5.1

ANSWER:

$$T(s) = \frac{s^3 + 1}{2s^4 + s^2 + 2s}$$

The complete solution is at www.wiley.com/college/nise

TryIt 5.1

Use the following MATLAB and Control System Toolbox statements to find the closed-loop transfer function of the system in Example 5.2 if all $G_i(s) = 1/(s + 1)$ and all $H_i(s) = 1/s$.

```
G1=tf(1,[1 1]);
G2=G1; G3=G1;
H1=tf(1,[1 0]);
H2=H1; H3=H1;
System=append...
(G1, G2, G3, H1, H2, H3);
input=1; output=3;
Q=[1 -4 0 0 0
  2 1 -5 0 0
  3 2 1 -5 -6
  4 2 0 0 0
  5 2 0 0 0
  6 3 0 0 0];
T=connect(System,...
Q, input, output);
T=tf(T); T=minreal(T)
```

In this section, we examined the equivalence of several block diagram configurations containing signals, systems, summing junctions, and pickoff points. These configurations were the cascade, parallel, and feedback forms. During block diagram reduction, we attempt to produce these easily recognized forms and then reduce the block diagram to a single transfer function. In the next section, we will examine some applications of block diagram reduction.

5.3 Analysis and Design of Feedback Systems

An immediate application of the principles of Section 5.2 is the analysis and design of feedback systems that reduce to second-order systems. Percent overshoot, settling time, peak time, and rise time can then be found from the equivalent transfer function.

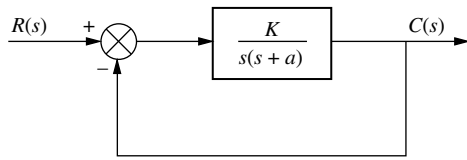


FIGURE 5.14 Second-order feedback control system

Consider the system shown in Figure 5.14, which can model a control system such as the antenna azimuth position control system. For example, the transfer function, $K/s(s+a)$, can model the amplifiers, motor, load, and gears. From Eq. (5.9), the closed-loop transfer function, $T(s)$, for this system is

$$T(s) = \frac{K}{s^2 + as + K} \quad (5.10)$$

where K models the amplifier gain, that is, the ratio of the output voltage to the input voltage. As K varies, the poles move through the three ranges of operation of a second-order system: overdamped, critically damped, and underdamped. For example, for K between 0 and $a^2/4$, the poles of the system are real and are located at

$$s_{1,2} = -\frac{a}{2} \pm \frac{\sqrt{a^2 - 4K}}{2} \quad (5.11)$$

As K increases, the poles move along the real axis, and the system remains overdamped until $K = a^2/4$. At that gain, or amplification, both poles are real and equal, and the system is critically damped.

For gains above $a^2/4$, the system is underdamped, with complex poles located at

$$s_{1,2} = -\frac{a}{2} \pm j\frac{\sqrt{4K - a^2}}{2} \quad (5.12)$$

Now as K increases, the real part remains constant and the imaginary part increases. Thus, the peak time decreases and the percent overshoot increases, while the settling time remains constant.

Let us look at two examples that apply the concepts to feedback control systems. In the first example, we determine a system's transient response. In the second example, we design the gain to meet a transient response requirement.

Example 5.3

Finding Transient Response

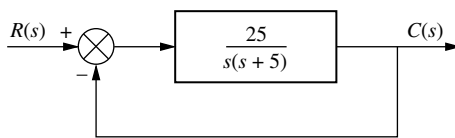


FIGURE 5.15 Feedback system for Example 5.3

PROBLEM: For the system shown in Figure 5.15, find the peak time, percent overshoot, and settling time.

SOLUTION: The closed-loop transfer function found from Eq. (5.9) is

$$T(s) = \frac{25}{s^2 + 5s + 25} \quad (5.13)$$

From Eq. (4.18),

$$\omega_n = \sqrt{25} = 5 \quad (5.14)$$

From Eq. (4.21),

$$2\zeta\omega_n = 5 \quad (5.15)$$

Substituting Eq. (5.14) into (5.15) and solving for ζ yields

$$\zeta = 0.5 \quad (5.16)$$

Using the values for ζ and ω_n along with Eqs (4.34), (4.38), and (4.42), we find respectively,

$$T_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} = 0.726 \text{ second} \quad (5.17)$$

$$\%OS = e^{-\zeta\pi/\sqrt{1-\zeta^2}} \times 100 = 16.303 \quad (5.18)$$

$$T_s = \frac{4}{\zeta\omega_n} = 1.6 \text{ seconds} \quad (5.19)$$

Students who are using MATLAB should now run ch5p2 in Appendix B. You will learn how to perform block diagram reduction followed by an evaluation of the closed-loop system's transient response by finding, T_p , $\%OS$, and T_s . Finally, you will learn how to use MATLAB to generate a closed-loop step response. This exercise uses MATLAB to do Example 5.3.

MATLAB
ML

MATLAB's Simulink provides an alternative method of simulating feedback systems to obtain the time response. Students who are performing the MATLAB exercises and want to explore the added capability of MATLAB's Simulink should now consult Appendix C. Example C.3 includes a discussion about, and an example of, the use of Simulink to simulate feedback systems with nonlinearities.

Simulink
SL

Example 5.4

Gain Design for Transient Response

PROBLEM: Design the value of gain, K , for the feedback control system of Figure 5.16 so that the system will respond with a 10% overshoot.

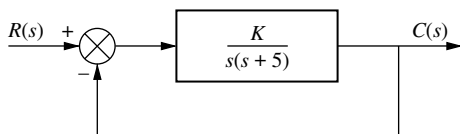


FIGURE 5.16 Feedback system for Example 5.4

SOLUTION: The closed-loop transfer function of the system is

$$T(s) = \frac{K}{s^2 + 5s + K} \quad (5.20)$$

From Eq. (5.20),

$$2\zeta\omega_n = 5 \quad (5.21)$$

and

$$\omega_n = \sqrt{K} \quad (5.22)$$

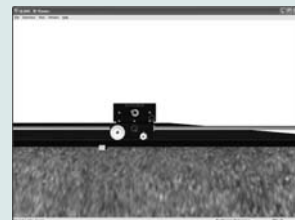
Thus,

$$\zeta = \frac{5}{2\sqrt{K}} \quad (5.23)$$

Since percent overshoot is a function only of ζ , Eq. (5.23) shows that the percent overshoot is a function of K .

Virtual Experiment 5.1 Position Control Gain Design

Put theory into practice designing the position control gain for the Quanser Linear Servo and simulating its closed-loop response in LabVIEW. This concept is used, for instance, to control a rover exploring the terrain of a planet.



Virtual experiment are found on WileyPLUS.

A 10% overshoot implies that $\zeta = 0.591$. Substituting this value for the damping ratio into Eq. (5.23) and solving for K yields

$$K = 17.9 \quad (5.24)$$

Although we are able to design for percent overshoot in this problem, we could not have selected settling time as a design criterion because, regardless of the value of K , the real parts, -2.5 , of the poles of Eq. (5.20) remain the same.

Skill-Assessment Exercise 5.2

PROBLEM: For a unity feedback control system with a forward-path transfer function $G(s) = \frac{16}{s(s+a)}$, design the value of a to yield a closed-loop step response that has 5% overshoot.

ANSWER:

$$a = 5.52$$

The complete solution is at www.wiley.com/college/nise.

TryIt 5.2

Use the following MATLAB and Control System Toolbox statements to find ζ , ω_n , %OS, T_s , T_p , and T_r for the closed-loop unity feedback system described in Skill-Assessment Exercise 5.2. Start with $a = 2$ and try some other values. A step response for the closed-loop system will also be produced.

```
a=2;
numg=16;
deng=poly([0 -a]);
G=tf(numg,deng);
T=feedback(G,1);
```

```
[numt,dent]=...
tfdata(T,'v');
wn=sqrt(dent/3)
z=dent(2)/(2*wn)
Ts=4/(z*wn)
Tp=pi/(wn*...
sqrt(1-z^2))
pos=exp(-z*pi*...
/sqrt(1-z^2))*100
Tr=(1.76*z^3+...
0.417*z^2+1.039*...
z+1)/wn
step(T)
```

5.4 Signal-Flow Graphs

Signal-flow graphs are an alternative to block diagrams. Unlike block diagrams, which consist of blocks, signals, summing junctions, and pickoff points, a signal-flow graph consists only of *branches*, which represent systems, and *nodes*, which represent signals. These elements are shown in Figure 5.17(a) and (b), respectively. A system is represented by a line with an arrow showing the direction of signal flow through the

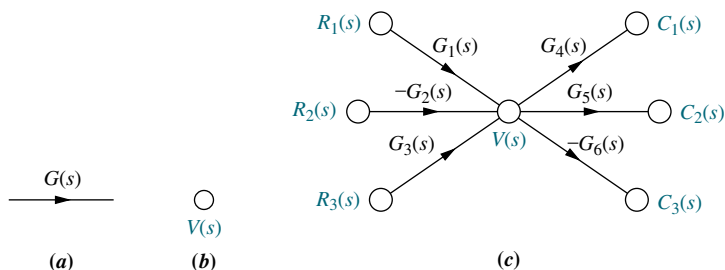


FIGURE 5.17 Signal-flow graph components: **a.** system; **b.** signal; **c.** interconnection of systems and signals